LLM trained bandit algorithms for improving Q-Commerce grocery swaps

Ioan Webber[1], Maharshi Dhada[1], Madalina Lupu[2], Matteo Giaretti[3], and Duncan McFarlane[1]

[1] Institute of Manufacturing, University of Cambridge, 17 Charles Babbage Road, CB3 0FS, United Kingdom
{iww22, mhd37, dm114}@cam.ac.uk
[2] Glovo, Carrer de Llull, 108, Sant Martí, 08005 Barcelona, Spain
madalina.lupu@glovoapp.com
[3] Independent Researcher
matteo.giaretti@gmail.com

Abstract. In the modern competitive marketplace, providing consumers with convenience, low costs, and a positive customer experience is essential. Online shopping at supermarkets via quick commerce (Q-commerce) has become increasingly common, presenting new challenges due to product catalogue sizes ranging from hundreds to tens of thousands. This makes stockouts or outdated product listings very common, requiring significant attention to maintain customer satisfaction by creating robust and scalable product swap systems. Product similarity is often used to suggest alternative product swaps, but large catalogues and limited customer feedback data make it difficult to monitor and change swap suggestions if they are not satisfactory. This paper explores methods for incorporating customer feedback to change product swap recommendations over time, by using *Thompson sampling multi-armed bandits* and *upper confidence bound (UCB) contextual bandits* with network-based experience sharing. A *multimodal LLM* is used to provide feedback for the process, enabling the generation of synthetic responses via human-like automation, in the absence of real-world data.

Keywords: Machine learning · Contextual bandit · Multi-armed bandit · Multimodal LLM · Recommender Systems · Q-Commerce

1 Introduction

1.1 Background and Motivation

Q-commerce, also known as 'quick commerce', is a type of e-commerce with an emphasis on deliveries with minimal lead time, often one hour or less[12]. It is used for on-demand food delivery and is increasingly expanding into the delivery of everyday products, including groceries, medicine, gifts, etc.. As Q-commerce is a relatively new value proposition[9], there is a continuous push to improve the customer experience by improving processes and developing new features.

This is especially true in the case of rapid grocery deliveries, which differentiates itself from fast-food orders by offering catalogues ranging from hundreds to tens of thousands of unique product variants, typically then delivered by mobile city bikes to the customer in as little as 30 minutes[31]. Inevitably, issues with stock-outs and missing products are very common, and can be as high as 48%[5]. To avoid the negative outcomes of poor customer experiences and complaints, it is critical to mitigate this issue by providing suitable alternatives in their place[13].

Working with a major European Q-Commerce provider, Glovo[1], the current approach for out-of-stock product replacement is based on compiling text and image similarity between products in the catalogue, and using this to inform on the most suitable replacements, somewhat similar to the approach described in [16]. While using text and image similarity can provide a suitable list of product swaps, its static nature prevents the incorporation of customer feedback. Creating the risk that poor product swaps can be repeated threatens customer satisfaction, so integrating a feedback mechanism is important to ensure the long-term suitability[14].

In this paper, reinforcement learning approaches are considered, focusing on multi-armed and contextual bandit methods[32] with similarity-based experience networked feedback sharing. Additionally, a method for providing synthetic feedback from a multi-modal LLM[22] is also considered in the absence of existing customer feedback. The structure is as follows: Section 2 is the literature review, Section 3 covers the methodology, including the design of the multi-armed bandit, contextual bandit and the use of the multi-modal LLM to provide feedback, Section 4 is the implementation of these algorithms, and Section 5 outlines the conclusions, with the limitations and scope for future work.

2 Literature Review

2.1 Bandit algorithms in recommendation systems

Reinforcement learning algorithms are used extensively within recommendation systems due to their ability to handle sequences of user interaction, deal with sparse feedback and balance the exploration and exploitation trade-off, which is critical for improving the quality of recommendations[23]. Bandit algorithms are a simplified version of reinforcement learning, which interact with their environment in an attempt to maximise reward via exploration and exploitation[28], differing from full reinforcement learning algorithms due to not being permitted to affect the state of the environment and the reward[2][32]. In the context of recommendation systems, this is not a requirement, and hence they have been extensively applied by companies including Amazon, Netflix, Spotify and more[7]. Within the class of bandit algorithms, the heuristics typically used to navigate the exploration-exploitation trade-off are ϵ-Greedy, Upper confidence bound (UCB), Thompson sampling[27], Softmax and EXP3[7]. ϵ-Greedy is the simplest of these, enforcing a fixed proportion of events which are chosen to be exploration. While the other heuristics dynamically adjust the trade-off based on the confidence in the options based on prior learning.

Contextual bandits are a more advanced option, leveraging additional information about the environment to enable more informed decisions about recommendations[28]. Using context allows the building of customer profiles and the sharing of information across product classes, improving the performance[20].

2.2 Bandit algorithms in e-commerce applications

E-commerce is one of the most challenging but important areas for recommendation systems. By supplying appropriate recommendations, it makes it simpler for customers to find relevant products, which in turn bolsters engagement and satisfaction[18]. This being the case, a rich body of literature on bandit algorithms has been developed to provide a diverse set of tools to support applications across a broad range of fields and scales of applications[19].

A foundational application of linear upper confidence bound contextual bandits was first demonstrated by Li et al. ([17]), capable of improving the click through of personalised news recommendations by 12.5% when compared to a context-free bandit. Since then, many types of contextual bandits have been applied in a variety of e-commerce settings[7], for example, authors such as Broden et al. ([4]) use Thompson sampling bandits to optimise item-to-item recommendations to the consumer, especially in cases in which user profiles have limited or non-existent information, which is often the case in online marketplaces. Contextual bandits are also increasingly employed in novel use-cases such as dynamic pricing[26], AB testing[25] and cold-start problems[11], which have all shown commercial value.

Within the remit of grocery product swaps, Ban et al. ([3]) propose a multinomial logistic regression model for presenting personalised recommendations for both substitutes and co-purchases for each customer within their online cart. Song ([29]) presents a Bayesian Bandit approach to providing personalised online coupon recommendations; however, to the awareness of the authors, there is no current literature which considers the application of bandit algorithms to product grocery swaps in e-commerce.

2.3 Large Language Models (LLMs) for feedback generation

One of the key difficulties with the introduction of a bandit recommendation system is known as the "cold-start" problem. This is faced where systems are unable to provide high-quality recommendations due to the lack of availability of training data[11][7][27][20]. One method for navigating this challenge is the inclusion of contextual factors[27][20], but another is to use synthetic data, derived from real data for pre-training or model evaluation[10][30].

The use of large language models (LLMs) in recommendation systems is a growing topic of interest[34], driven by the high rate of development and impressive capabilities for reasoning and generalisation. They have been proposed for use as chatbots[6], attribute extraction[8] and personalised refinement of product page listings[15]. While they cannot currently surpass the human capacity

to undergo complex decision-making processes based on past experiences, they offer a cost-effective and time-efficient alternative[24].

3 Methodology

3.1 Overview

This section outlines the methodology to support the addition of feedback to improve product swap recommendations in Q-Commerce. Recognising the limits of the existing system, which compiles static product similarity scores, we incorporate adaptive algorithms capable of learning with feedback to adjust recommendations.

Multi-armed and contextual bandit approaches are used, which can effectively integrate with existing knowledge of the products, and use networked feedback to share information within similar product clusters effectively. Additionally, to address the real-world constraint of customer feedback data, multimodal large language models are used to generate synthetic feedback. Combined, these methods present a scalable, responsive and data-efficient product swap recommendation system.

3.2 Multi-armed Bandit

Model Definition The multi-armed bandit problem is typically described by considering several slot machines, or actions that can be taken. After each choice, you receive a reward from an unknown probability distribution that depends on the action you selected[32]. The multi-armed bandit algorithm presented here uses a Thompson sampling[28], leveraging Beta distributions to model the expected reward for every action. Following each action, the reward is incorporated back into the system, and the distributions are updated to reflect the additional knowledge gained about the environment. The equations provided are based on the literature of [32] and [28], but developed to tailor to this application.

The distributions are set with P defined as the set of all products, S_{ij} represents the similarity between products i and j and for each product pairing (i, j) there is a Beta distribution with parameters $(\alpha_{ij}, \beta_{ij})$. The Beta distribution is parametrised as $Beta(\alpha, \beta) \propto x^{\alpha-1}(1-x)^{\beta-1}$.

To mitigate the cold-start problem, the beta distribution parameters are derived from the existing similarity scores. The initial beta distribution parameters are derived by using the equations:

$$\alpha_{i,j} = S_{i,j} \times \lambda + 1 \tag{1}$$

$$\beta_{i,j} = (1 - S_{i,j}) \times \lambda + 1 \tag{2}$$

In equations 1 and 2, λ determines the weighting of the impact of the similarity scores on the initial distribution. A smaller value of λ makes the algorithm less reliant on the initial similarity, favouring a more explorative approach. The

mean of the Beta distribution is given by $\mu = \frac{\alpha_{ij}}{\alpha_{ij}+\beta_{ij}}$, meaning that, provided that $S_{ij} \times \lambda \gg 1$, in the absence of feedback, the model will behave very similarly to that of one based entirely on the similarity scores.

Recommendations The purpose of this multi-armed bandit algorithm is to suggest several products to the order picker, which are all considered good matches for the original out-of-stock product. The input variables are defined as the out-of-stock product p, the total available products N, and the number of products to show to the picker n. The set of similar products is defined by using a dynamic threshold T:

$$T = \begin{cases} median(S_p) & \text{if N} > 7000 \\ percentile_{25}(S_p) & \text{otherwise} \end{cases} \tag{3}$$

Where each product $p_i \in P_p$, the set of similar products is defined by (4), and the corresponding similarity scores (5) are given by:

$$P_p = \{p_i | S_{p,i} > T \ \wedge \ p_i \neq p\} \tag{4}$$

$$S_p = \{S_{p,i} | p_i \in P_p\} \tag{5}$$

For each similar product p_i, a random sample is then taken from the beta distribution, given by the equation:

$$\chi = \{(p_i, X_i) | p_i \in P_p \text{ and } X_i \sim Beta(\alpha_i, \beta_i)\} \tag{6}$$

The picks are then shown to the picker, are then given by:

$$Top_n(p) = \{p_j | (p_j, X_j) \in sort_{desc}(\chi)[1:n]\} \tag{7}$$

Feedback Updates The update function adjusts the distributions based on feedback, updating the posterior distributions and enabling changes to the recommendations. Due to limited feedback on specific products in large catalogues, similarity-based experience sharing propagates feedback indirectly through the network, speeding up the convergence of similar product clusters.

For the out-of-stock product p, reviewed by customer r and Reward R, the direct feedback is given by 8 for positive feedback and 9 for negative.

$$\alpha_{p,r} \leftarrow \alpha_{p,r} + \gamma \times R \tag{8}$$

$$\beta_{p,r} \leftarrow \beta_{p,r} + \gamma \times R \tag{9}$$

γ is the reward rate and can be adjusted to influence the reactivity to feedback. Indirect feedback is propagated in proportion to the square of the similarity to the reviewed product. For another product, $q \in P_p$, $S_{q,r}$ represents the similarity between that product q and the reviewed product p. The exponential favours spreading feedback amongst most similar products, typically those which are the same brand or style.

$$\alpha_{q,r} \leftarrow \alpha_{q,r} + \gamma \times S_{q,p}^2 \times R \tag{10}$$

$$\beta_{q,r} \leftarrow \beta_{q,r} + \gamma \times S_{q,p}^2 \times R \tag{11}$$

3.3 Contextual Bandit Model

Model Definition The contextual bandit model extends the multi-armed bandit via the incorporation of additional information, or 'context', about the products[32]. For example, a customer may dislike a product swap which provides a smaller quantity or a higher price; these features cannot be captured by a multi-armed bandit but can be included as useful information in the contextual bandit model. The approach here uses a Linear Upper Confidence Bound (LinUCB) contextual bandit[32], which can be programmed with distinct contextual variables which are trackable over time to understand the influence of feedback. It is a widely used approach, effective at balancing the exploration-exploitation trade-off, and incorporating uncertainty into the reward estimation, making it suitable for starting scenarios with limited data[20].

The algorithm is set up using the set of available products P, with an out-of-stock product $p_t \in P$ at each time step t and some customer context $c_t \in \mathbb{R}^{d-q}$. Where d is the dimensionality of the context vector and q is the number of pre-computed context variables. The action a is to select a replacement for the out-of-stock product $a_{p,t} \in P$, with the algorithm receiving a reward r_t associated with the action and context.

A feature tracking matrix A of dimension d^2 and a cumulative rewards vector b of length d are also created. The weights vector θ, of length d, stores the relative importance of each contextual variable to the decision. This is made up of a weighting, θ, between the local and global variables, associated with a single customer and the whole customer pool. This is given by 12, where δ is the feedback parameter $[0, 1]$ that determines the balance.

$$\theta = \delta\theta_{Local} + (1 - \delta)\theta_{Global} \tag{12}$$

Recommendation The expected payoff of an action, $Q_t(a)$, is given below, with the strategy for choosing the action A_t^{top-n}, are given below. α is the exploration parameter that balances the exploration-exploitation trade-off.

$$Q_t(a) = \mathbb{E}[r_t|x_t, a_t] = x_{p,a}^T \theta_{p,a} \tag{13}$$

$$A_t^{top-n} = Top - n_{a \in A} \left(Q_t(a) + \alpha \sqrt{x_t^T A_a^{-1} x_t} \right) \tag{14}$$

Updates The local updates, $A_{local}, b_{local}, \theta_{local}$, are generated using recursive linear algebra, using the feature matrix and the cumulative reward vector initialised previously, shown in equations 15 - 17.

$$A_{Local} \leftarrow A_{Local} + xx^T \tag{15}$$

$$b_{Local} \leftarrow b_{Local} + rx \tag{16}$$

$$\theta_{Local} \leftarrow A_{Local}^{-1} b_{Local} \tag{17}$$

The updates to local and global variables are processed separately. The global updates are computed similarly, to (15) - (17), but with b_{Global} containing an additional decay factor ϵ, which decreases the value of the stored variables over time, enabling the global variables to respond to trends more effectively, i.e.

$$b_{Global} \leftarrow (1 - \epsilon) \times b_{Global} + rx \tag{18}$$

The updates are spread to the networks of similar products via a similar network sharing to that in section 3.2. In these cases, the indirect reward can be propagated via equation 19, where S_{rq} is the similarity between the reviewed product r and the similar product p which is receiving the networked feedback.

$$b \leftarrow b + S_{r,q}^2 \times rx \tag{19}$$

3.4 Synthetic feedback from LLMs

Background A key challenge with creating reinforcement learning recommender systems is the requirement for data feedback to enable improvements in the recommendations. In a database of 7000 products, there are 24496500 possible product swaps, making it costly and time-consuming to check each of these manually. An alternative is to use synthetic data, which, while potentially lower quality, is abundant and cheap to produce.

There is increasing interest in the application of LLMs in recommender systems, due to their language understanding, generalisation capabilities and the ability to reason[34][6]. Their ability to automate human-mimicked reasoning represents an exciting opportunity for feedback generation, as it allows us to address the difficulties of achieving thousands of product comparisons. There are no specific language or image models for the specific application of comparing grocery products; however, there are models which can handle complex image and text data, which should be trained on similar types of image and text data. Multimodal language models can process both in tandem to generate a similarity score between the options. The model used here is CLIP, released by OpenAI in 2021[22]. It was jointly trained by using an image encoder and a text encoder to predict pairings using a dataset of 400 million image-text pairs.

Implementation CLIP works by decomposing the text and image prompts into two 768-dimensional vectors, which can be concatenated and compared between products using cosine similarity, via the equation:

$$\text{Cosine similarity } = \frac{A \cdot B}{\|A\|\|B\|} \tag{20}$$

The LLM text and image embeddings are used as they are an efficient representation of high-dimensional data in a compact, comparable format[15]. Embeddings preserve essential semantic information, enabling complex comparisons using relatively little computational overhead[33], making them ideal for comparisons in large databases.

Due to the residual similarity between the prompts and image formatting, the average background cosine similarity (μ) and standard deviation (σ) are included to ensure the similarity figure remains a useful quantity. Feedback can be

$$\text{Standardised similarity} = \frac{\text{Cos sim.} - \mu}{\sigma} \tag{21}$$

In the next section, we demonstrate the application of the approach introduced here to retail examples.

4 Results

4.1 Data Sources and Preprocessing

The implementation of these approaches uses information from several supermarket catalogues, including the product names, weight, brand and an image. Additionally, a similarity matrix for each supermarket, using a combination of the image and text similarities to create a similarity score between 0 and 1 for every product combination, had already been created. These were produced separately by using Jaccard similarity[21] for the text and a pre-trained image analysis model to create high-level embeddings, which can be compared using cosine similarity.

The existing similarity model was verified by taking a sub-sample of 180 products each from three different supermarket chains located in separate countries. From this subset, 2058 potential product swaps were suggested to the supermarket managers, in which they identified 91.1% as suitable substitutes, 4.9% as poor substitutes and 4.0% as unsure.

4.2 Multi-armed Bandit

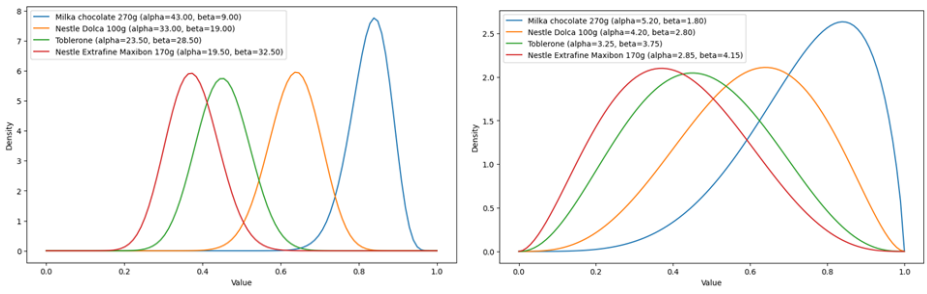
Initialisation The implementation of the Thompson Sampling multi-armed bandit for this application is used as a basis to test the scalability in a catalogue of approximately 10000 products and trial the effects of network sharing.

Table 1 shows the similarity scores between a selection of five chocolate bars from the catalogue, and from this, the means of the Beta distributions in Figure

1 are derived, using equations 1 and 2. The inclusion of the λ term enables the system to initially favour exploration or exploitation, depending on the level of trust placed on the derived similarity scores. This can be seen in the comparison between the initialised Beta distributions in Figure 1.

Index	**Milka chocolate 125g™**	**Milka chocolate 270g™**	**Nestle Dolce 100g™**	**Toblerone 100g™**	**Nestle Maxibon 170g™**
Milka chocolate 125g™	1	0.84	0.64	0.45	0.37
Milka chocolate 270g™	0.84	1	0.59	0.51	0.32
Nestle Dolce 100g™	0.64	0.59	1	0.48	0.48
Toblerone 100g™	0.45	0.51	0.48	1	0.35
Nestle Extrafine 170g™	0.37	0.32	0.48	0.35	1

Table 1: Example Comparison of Similar Chocolate Products



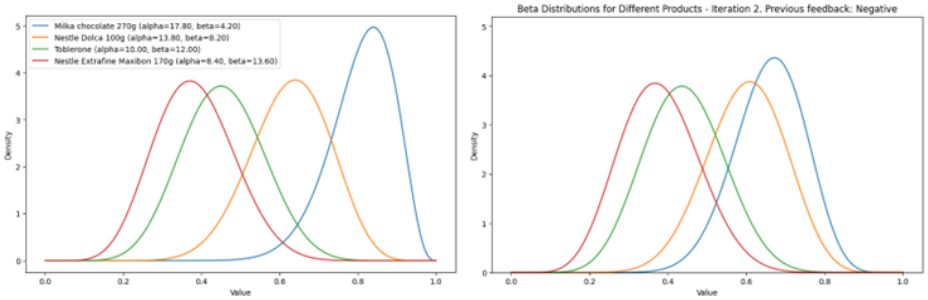
(a) Favouring Exploitation (b) Favouring Exploration

Fig. 1: Initialised Beta Distributions For Multi-armed Bandit

Feedback Integration There are three types of feedback, which are considered: customer feedback, picker feedback and synthetic feedback. Each can be weighted differently depending on feedback is positive or negative, and their relative assigned importance. A demonstration of negative customer feedback on the beta distributions is shown in Figure 2, with negative feedback applied to the 'Milka Chocolate 270g', causing the expected reward to decrease.

The effect of network sharing was also trialled alongside positive synthetic feedback for around 20% of the items in the dataset. This had the overall effect of increasing the total number of new product suggestions provided to the picker by around 5% across all the products, compared to when it is not used. The effect of this can be altered by changing the feedback parameters, but this demonstrates

its capacity to accelerate the rate at which similar product clusters can be shown to the picker.



(a) Prior to Feedback (b) After Negative Feedback Received

Fig. 2: Change in Beta Distributions After Negative Feedback to First Item

4.3 Contextual Bandit

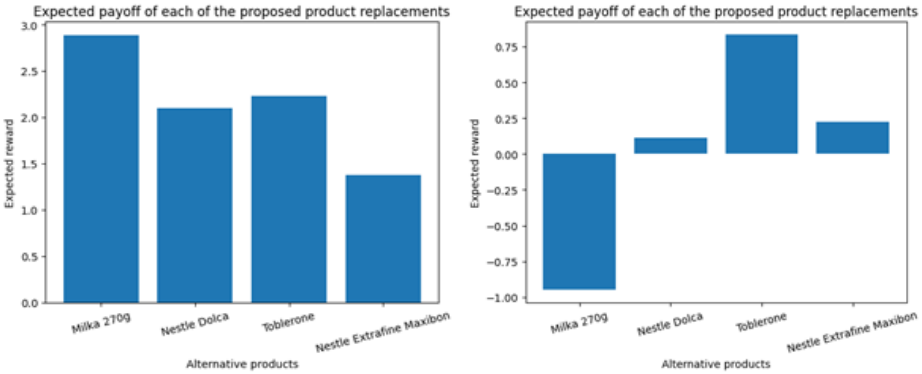
Initialisation The contextual variables used in this implementation are in (22):

$$\text{Context} \quad = x = \begin{bmatrix} \text{Similarity score} \\ \text{cost ratio} \\ \text{weight ratio} \\ \text{avg cost/weight} \\ \text{avg spend} \\ \text{avg weight} \\ \text{branded proportion} \end{bmatrix} \tag{22}$$

The variables are chosen to reflect expected categories likely to create negative customer experiences, but it should be noted that they are not exhaustive. They are made from a combination of pre-computed product characteristics relating to the out-of-stock product ($x_{1:3}$) and customer preferences ($x_{4:7}$), enabling both individual (local) and global context to shape the decision process.

Feedback Integration The results of feedback integration in the contextual bandit model are presented in Figure 3. The leftmost plot shows the initial expected payoff by selecting each option of the alternative products, before feedback is received. In this case, the optimal selection is to choose 'Milka 270g'.

Should this option be provided and the feedback be negative, the expected payoff for the next round becomes that shown by figure 3b. The model has reacted and has changed the expected payoff of the Milka bar accordingly, as well as changing the payoffs of the other products, due to the influence of the feedback context. Given this, the next best recommendation is the 'Toblerone', as the algorithm favours its differentiated contextual features.



(a) Prior to Feedback (b) After Negative Feedback Received

Fig. 3: Expected Reward after Negative Feedback Received

By using a linear upper confidence bound contextual bandit approach, compared to deep neural networks, for example, there is the additional ability to understand the specific changes in the rewards vector once feedback is received. This is shown in figure 4, by the context vector placing a negative weight on products which have a higher price than the original. This provides direct insight into the specific contextual properties which determine effective product swaps.

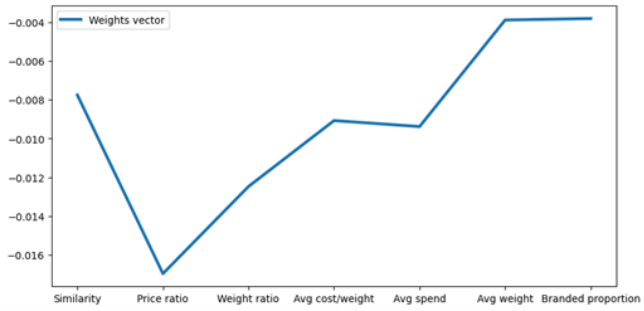


Fig. 4: Change to Weights Vector in Response to Negative Feedback

4.4 Synthetic Feedback Comparison

For the generation of synthetic feedback data using the multimodal LLM, introduced in section 3.4, the comparison included a standardised image of the product before a white background, with the prompt: “The product name is:

Product name. It costs Price in local currency. The product brand is: Brand.", where bracketed values are auto-filled with information from the database.

The cosine similarity, calculated using (20), has a range [-1,1], but due to the residual similarities in the formatting of the prompt and image layout, this is not evenly distributed about a mean of 0. Therefore, a random sample of 1000 product swap suggestions can be taken from the database to find the background average similarity (μ) and standard deviation (σ).

Comparing existing non-zero similarity scores with the LLM similarities results in Figure 5. To ensure confidence in the synthetic feedback used, a 3σ filtering threshold is taken to the CLIP-based similarity scores. Given the original similarity engine was verified with 91.1% accuracy, and shows strong agreement with the synthetic LLM similarity. This subset is taken to represent the subset of high-quality synthetic feedback for the model reinforcement.

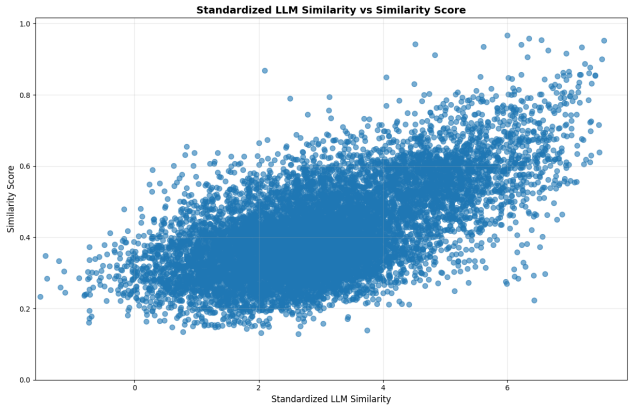


Fig. 5: Standardised CLIP similarity scores compared to existing similarity scores

When the 3σ threshold is taken in a real retail dataset, it corresponds to around 20% of the set of all the identified product swaps. Using this synthetic feedback to update the model weights decreased the number of new options shown to the picker by 4%, demonstrating that this improves the certainty of the model, although it is worth noting that the effect of this can be adjusted by varying the input parameters.

In summary, the results in this section have shown that using a light-weight LLM with open source weights to compare product availability produces comparable results to the purpose-built text and image similarity engine. This comparison can be automated simply, and open alternative routes for collecting synthetic feedback.

5 Conclusions

The purpose of this research has been to create a hybrid model, based on an existing similarity engine, capable of responding to user feedback to improve its recommendations over time. Unfortunately, there was no customer data available in which to trial the model's performance; however, it is possible to demonstrate the capacity of the model to learn and change parameters over time. By using a multi-armed or contextual bandit model, it is much simpler to track the changes in parameter weights over time and in response to feedback when compared to neural networks, which are a popular alternative.

From the business perspective, these improvements should enable pickers to spend less time reviewing substitute suggestions and enhance customer satisfaction which are key performance drivers in Q-Commerce.

Networked experience sharing creates the ability to make the most of feedback by propagating it through similar networks of products. This makes programs more effective at utilising an ideally small number of customer responses, to make the change as impactful as possible. In practice, this appeared to have the effect of introducing around 5% more suggestions when introduced alongside the LLM feedback, based on our training parameters. This could also be extended to second-degree, or further, to perhaps introduce products which may not have been initially considered similar to the initial out-of-stock product, as well.

The inclusion of the Multimodal LLM to provide feedback into the system appeared to work well, with the average LLM standardised similarity score produced compared to the existing text and image similarity engines being 3σ from the mean of the dataset as a whole. Applying this feedback as updates to the multi-armed bandit model showed that it decreased the number of new items shown to pickers by 4%, demonstrating that it used the positive feedback to increase the certainty of the model.

5.1 Limitations and Future Work

It has not been possible to implement these models to receive live data, which makes it difficult to know the extent to which they will be effective in practice. Contextual factors have been included based on the assumption that these will be impactful in understanding what constitutes an effective product swap suggestion. However, without using customer data, the assumptions of linearity and contextual variables may not hold, and it might instead be more effective to implement it as a neural contextual bandit or other non-linear model.

Also, the CLIP Multimodal LLM[22] used here was developed in 2021, and while useful as a lightweight and open-source model, it has a very limited number of input tokens and parameters compared to updated versions. It would also be interesting to investigate further trialling different prompts using newer models, to determine the extent to which there is an understanding of specific contextual factors relating to products as well.

References

1. Home - Glovo Corporate Site (Jun 2025), https://about.glovoapp.com
2. Afsar, M.M., Crump, T., Far, B.: Reinforcement Learning based Recommender Systems: A Survey. ACM Computing Surveys **55**(7), 1–38 (Dec 2022). https://doi.org/10.1145/3543846
3. Ban, G.Y., BENBITOUR, M.H., Chen, B.: Selling personalized substitutes and co-purchases in online grocery retail. Available at SSRN (2024)
4. Brodén, B., Hammar, M., Nilsson, B.J., Paraschakis, D.: A bandit-based ensemble framework for exploration/exploitation of diverse recommendation components: An experimental study within e-commerce. ACM Transactions on Interactive Intelligent Systems (TiiS) **10**(1), 1–32 (2019)
5. Chan, S.: 13 of the worst supermarket substitution fails - revealed. Which? (Mar 2017), https://www.which.co.uk/news/article/13-of-the-worst-supermarket-substitution-fails-revealed-a3fxh6s6s2yd
6. Chang, T.J., Lin, L.H.M., Tsai, R.T.H.: Conversational product recommendation using llm. In: 2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB). pp. 340–343 (2024). https://doi.org/10.1109/ICEIB61477.2024.10602608
7. Elena, G., Milos, K., Eugene, I.: Survey of multiarmed bandit algorithms applied to recommendation systems. International Journal of Open Information Technologies **9**(4), 12–27 (2021)
8. Fang, C., Li, X., Fan, Z., Xu, J., Nag, K., Korpeoglu, E., Kumar, S., Achan, K.: Llm-ensemble: Optimal large language model ensemble method for e-commerce product attribute value extraction. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 2910–2914. SIGIR '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3626772.3661357, https://doi.org/10.1145/3626772.3661357
9. Fornari, E., Negri, F., Iuffmann Ghezzi, A.: The wheel of retailing is still spinning: promises, compromises, and pitfalls of quick-commerce. In: NEXT GENERATION MARKETING. People, Planet, Place: cooperation & shared value for a new era of critical marketing (Proceedings), pp. 1–12. N/A, Italy (2022), https://publicatt.unicatt.it/handle/10807/224629
10. FRABETTI, A.: Exploring the employment of synthetic data in recommendation systems (2023)
11. Gope, J., Jain, S.K.: A survey on solving cold start problem in recommender systems. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). pp. 133–138 (2017). https://doi.org/10.1109/CCAA.2017.8229786
12. Harter, A., Stich, L., Spann, M.: The effect of delivery time on repurchase behavior in quick commerce. Journal of Service Research **28**(2), 211–227 (2025). https://doi.org/10.1177/10946705241236961, https://doi.org/10.1177/10946705241236961
13. Hoang, D., Breugelmans, E.: "Sorry, the product you ordered is out of stock": Effects of substitution policy in online grocery retailing. Journal of Retailing **99**(1), 26–45 (Mar 2023). https://doi.org/10.1016/j.jretai.2022.06.006
14. Hofstetter, J.S., Gruen, T., Ehrenthal, J.: Value attenuation and retail out-of-stocks: A service-dominant logic perspective. International Journal of Physical Distribution Logistics Management **44** (03 2014). https://doi.org/10.1108/IJPDLM-02-2013-0028

15. Kathiriya, S., Mullapudi, M., Karangara, R.: Optimizing ecommerce listing: Llm based description and keyword generation from multimodal data. International Journal of Science and Research (IJSR) **12** (10 2023). https://doi.org/10.21275/SR24304113521
16. Kerek, H.: Product Similarity Matching for Food Retail using Machine Learning (2020), https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1431623&dswid=-7406
17. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: ACM Other conferences, pp. 661–670. Association for Computing Machinery, New York, NY, USA (Apr 2010). https://doi.org/10.1145/1772690.1772758
18. Lin, Z.: An empirical investigation of user and system recommendations in e-commerce. Decision Support Systems **68**, 111–124 (Dec 2014). https://doi.org/10.1016/j.dss.2014.10.003
19. Liu, Y., Li, L.: A map of bandits for e-commerce. arXiv preprint arXiv:2107.00680 (2021)
20. Pilani, A., Mathur, K., Agrawald, H., Chandola, D., Tikkiwal, V.A., and, A.K.: Contextual bandit approach-based recommendation system for personalized web-based services. Applied Artificial Intelligence **35**(7), 489–504 (2021). https://doi.org/10.1080/08839514.2021.1883855, https://doi.org/10.1080/08839514.2021.1883855
21. Pradhan, N., Gyanchandani, M., Wadhvani, R., et al.: A review on text similarity technique used in ir and its application. International Journal of Computer Applications **120**(9), 29–34 (2015)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
23. Rossiiev, O.D., Shapovalova, N.N., Rybalchenko, O.H., Striuk, A.M.: A comprehensive survey on reinforcement learning-based recommender systems: State-of-the-art, challenges, and future perspectives. In: CEUR Workshop Proceedings. pp. 428–440 (2025)
24. Roumeliotis, K.I., Tselikas, N.D., Nasiopoulos, D.K.: Precision-driven product recommendation software: Unsupervised models, evaluated by gpt-4 llm for enhanced recommender systems. Software **3**(1), 62–80 (2024). https://doi.org/10.3390/software3010004, https://www.mdpi.com/2674-113X/3/1/4
25. Satyal, S., Weber, I., Paik, H.y., Di Ciccio, C., Mendling, J.: AB Testing for Process Versions with Contextual Multi-armed Bandit Algorithms. In: Advanced Information Systems Engineering, pp. 19–34. Springer, Cham, Switzerland (May 2018). https://doi.org/10.1007/978-3-319-91563-0_2
26. Sethuraman, S., Maheswari, G.U., Thombre, S., Kumar, S., Patel, V., Ramanan, S.: Encode: Ensemble contextual bandits in big data settings-a case study in e-commerce dynamic pricing. In: 2023 IEEE International Conference on Big Data (BigData). pp. 5372–5381. IEEE (2023)
27. Silva, N., Werneck, H., Silva, T., Pereira, A.C.M., Rocha, L.: Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions. Expert Systems with Applications **197**, 116669 (Jul 2022). https://doi.org/10.1016/j.eswa.2022.116669
28. Slivkins, A.: Introduction to Multi-Armed Bandits. Foundations and Trends in Machine Learning Ser. ; v. 38 (2019)

29. Song, X., et al.: A Bayesian bandit approach to personalized online coupon recommendations. Ph.D. thesis, Massachusetts Institute of Technology (2016)
30. Stavinova, E., Grigorievskiy, A., Volodkevich, A., Chunaev, P., Bochenina, K., Bugaychenko, D.: Synthetic data-based simulators for recommender systems: A survey. arXiv preprint arXiv:2206.11338 (2022)
31. Stojanov, M.: Q-commerce – the next generation e-commerce **1**, 17–34 (05 2022)
32. Sutton, R.S.: Reinforcement learning : an introduction / Richard S. Sutton and Andrew G. Barto. Adaptive computation and machine learning, second edition. edn. (2018)
33. Tao, C., Shen, T., Gao, S., Zhang, J., Li, Z., Tao, Z., Ma, S.: Llms are also effective embedding models: An in-depth overview. arXiv preprint arXiv:2412.12591 (2024)
34. Wang, Q., Li, J., Wang, S., Xing, Q., Niu, R., Kong, H., Li, R., Long, G., Chang, Y., Zhang, C.: Towards next-generation llm-based recommender systems: A survey and beyond. arXiv preprint arXiv:2410.19744 (2024)